

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

WHAT IS CLAIMED IS:

1           1.       An article of manufacture including program logic for performing  
2 configuration checking of a network, wherein the program logic causes operations to be  
3 performed, the operations comprising:  
4           scanning a network data store for at least one transaction;  
5           generating at least one event for said transaction;  
6           associating at least one configuration policy with said event;  
7           comparing said configuration policy with configuration data associated with said  
8 event; and  
9           determining whether said configuration policy has been violated based on the  
10 comparison.

1           2.       The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           generating at least one trigger for said event; and  
4           associating said configuration policy with said trigger.

1           3.       The article of manufacture of claim 1, wherein said configuration policy is  
2 retrieved from a local policy data store.

1           4.       The article of manufacture of claim 3, wherein said configuration policy in  
2 the local policy data store is automatically updated with a configuration policy in a  
3 remote data store.

1           5.       The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           receiving a hypothetical network scenario;  
4           generating at least one transaction based on the hypothetical network scenario;  
5           populating the network data store with configuration data for said transaction; and  
6           after determining whether said configuration policy has been violated based on  
7 the comparison, rolling back said transaction.

1           6.       The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           receiving a request to perform configuration checking on an existing network  
4 configuration.

1           7.       The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           when said configuration policy has been violated, performing an action specified  
4 in that configuration policy.

1           8.       The article of manufacture of claim 7, wherein the action is at least one of  
2 logging an indication that the configuration policy has been generated, generating at least  
3 one policy violation event, sending a notification, and highlighting a network topology  
4 viewer that graphically depicts the network.

1           9.       The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           when said configuration policy has been violated,

4                   accessing a solution in a knowledge data store; and  
5                   applying the solution so that said configuration policy is not violated.

1           10.    The article of manufacture of claim 1, wherein the operations further  
2 comprise:  
3           when said configuration policy has been violated,  
4                   determining that a component in the network is able to provide a solution;  
5 and  
6                   allowing the component to apply the solution so that said configuration  
7 policy is not violated.

1           11.    The article of manufacture of claim 1, wherein the operations for  
2 determining whether said configuration policy has been violated further comprise at least  
3 one of identifying incompatibilities between components in the network, performance  
4 issues, and availability issues.

1           12.    An article of manufacture including program logic for performing  
2 proactive configuration checking of a network, wherein the program logic causes  
3 operations to be performed, the operations comprising:  
4           receiving a hypothetical network scenario;  
5           generating at least one transaction based on said hypothetical network scenario;  
6           populating a network data store with configuration data for said transaction;  
7           generating at least one event for said transaction using a mapping of events to  
8 transactions; and  
9           using configuration data associated with said event to determine whether a  
10 configuration policy has been violated.

1           13.    The article of manufacture of claim 12, wherein the operations further  
2 comprise:  
3           rolling back said transaction by removing the configuration data for said  
4 transaction from the network data store.

1           14.    An article of manufacture including program logic for performing reactive  
2 configuration checking of a network, wherein the program logic causes operations to be  
3 performed, the operations comprising:  
4           receiving a request to perform configuration checking on an existing network  
5 configuration;  
6           scanning a network data store for at least one transaction;  
7           generating at least one event for said transaction using a mapping of events to  
8 transactions; and  
9           using configuration data associated with said event to determine whether a  
10 configuration policy has been violated.

1           15.    The article of manufacture of claim 14, wherein the operations further  
2 comprise:  
3           when said configuration policy has been violated, automatically correcting the  
4 violation.

1           16.    An article of manufacture including program logic for correcting a  
2 configuration problem, wherein the program logic causes operations to be performed, the  
3 operations comprising:  
4           discovering the configuration problem;

5           determining whether there is at least one solution for the configuration problem in  
6 a knowledge data store;  
7           when it is determined that there is at least one solution in the knowledge data  
8 store, automatically selecting a solution to solve the configuration problem;  
9           when said solution can be automatically applied, automatically applying said  
10 solution; and  
11          when said solution cannot be automatically applied, notifying a user.

1           17.    The article of manufacture of claim 16, wherein operations for detecting  
2 the configuration problem further comprise:  
3           periodically interrogating said component in the network for data; and  
4           determining whether there is a configuration problem based on the interrogation.

1           18.    The article of manufacture of claim 16, wherein operations for detecting  
2 the configuration problem further comprise:  
3           receiving at least one report from at least one component in the network; and  
4           determining whether there is a configuration problem based on the report.

1           19.    The article of manufacture of claim 16, wherein the operations further  
2 comprise:  
3           when a component in the network identifies a solution, allowing the component to  
4 automatically apply the solution.

1           20.    The article of manufacture of claim 16, wherein the configuration problem  
2 is at least one of a network configuration problem and a storage configuration problem.

1        21.    A system for performing configuration checking of a network, comprising:  
2        a processor;  
3        a computer readable medium accessible to the processor; and  
4        program logic including code capable of causing the processor to perform:  
5               scanning a network data store for at least one transaction;  
6               generating at least one event for said transaction;  
7               associating at least one configuration policy with said event;  
8               comparing the said configuration policy with configuration data associated  
9 with said event; and  
10               determining whether said configuration policy has been violated based on  
11 the comparison.

1        22.    The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:  
3               generating at least one trigger for said event; and  
4               associating said configuration policy with said trigger.

1        23.    The system of claim 21, wherein said configuration policy is retrieved  
2 from a local policy data store.

1        24.    The system of claim 23, wherein said configuration policy in the local  
2 policy data store is automatically updated with a configuration policy in a remote data  
3 store.

1        25.    The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:

3 receiving a hypothetical network scenario;  
4 generating at least one transaction based on the hypothetical network scenario;  
5 populating the network data store with configuration data for said transaction; and  
6 after determining whether said configuration policy has been violated based on  
7 the comparison, rolling back said transaction.

1 26. The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:  
3 receiving a request to perform configuration checking on an existing network  
4 configuration.

1 27. The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:  
3 when said configuration policy has been violated, performing an action specified  
4 in that configuration policy.

1 28. The system of claim 27, wherein the action is at least one of logging an  
2 indication that the configuration policy has been generated, generating at least one policy  
3 violation event, sending a notification, and highlighting a network topology viewer that  
4 graphically depicts the network.

1 29. The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:  
3 when said configuration policy has been violated,  
4 accessing a solution in a knowledge data store; and  
5 applying the solution so that said configuration policy is not violated.



1           30.    The system of claim 21, wherein the code is capable of causing the  
2 processor to further perform:  
3           when said configuration policy has been violated,  
4                 determining that a component in the network is able to provide a solution;  
5 and  
6                 allowing the component to apply the solution so that said configuration  
7 policy is not violated.

1           31.    The system of claim 21, wherein the code for determining whether said  
2 configuration policy has been violated is capable of causing the processor to further  
3 perform at least one of identifying incompatibilities between components in the network,  
4 performance issues, and availability issues.

1           32.    A system for performing proactive configuration checking of a network,  
2 comprising:  
3           a processor;  
4           a computer readable medium accessible to the processor; and  
5           program logic including code capable of causing the processor to perform:  
6                 receiving a hypothetical network scenario;  
7                 generating at least one transaction based on said hypothetical network  
8 scenario;  
9                 populating a network data store with configuration data for said  
10 transaction;  
11                 generating at least one event for said transaction using a mapping of  
12 events to transactions; and

13                    using configuration data associated with said event to determine whether a  
14 configuration policy has been violated.

1            33.    The system of claim 32, wherein the code is capable of causing the  
2 processor to further perform:  
3            rolling back said transaction by removing the configuration data for said  
4 transaction from the network data store.

1            34.    A system for performing reactive configuration checking of a network,  
2 comprising:  
3            a processor;  
4            a computer readable medium accessible to the processor; and  
5            program logic including code capable of causing the processor to perform:  
6            receiving a request to perform configuration checking on an existing  
7 network configuration;  
8            scanning a network data store for at least one transaction;  
9            generating at least one event for said transaction using a mapping of  
10 events to transactions; and  
11            using configuration data associated with said event to determine whether a  
12 configuration policy has been violated.

1            35.    The system of claim 34, wherein the code is capable of causing the  
2 processor to further perform:  
3            when said configuration policy has been violated, automatically correcting the  
4 violation.

1        36.    A system for correcting a configuration problem, comprising:  
2        a processor;  
3        a computer readable medium accessible to the processor; and  
4        program logic including code capable of causing the processor to perform:  
5            discovering the configuration problem;  
6            determining whether there is at least one solution for the configuration  
7 problem in a knowledge data store;  
8            when it is determined that there is at least one solution in the knowledge  
9 data store, automatically selecting a solution to solve the configuration problem;  
10          when said solution can be automatically applied, automatically applying  
11 said solution; and  
12          when said solution cannot be automatically applied, notifying a user.

1        37.    The system of claim 36, wherein the code for detecting the configuration  
2 problem is capable of causing the processor to further perform:  
3        periodically interrogating said component in the network for data; and  
4        determining whether there is a configuration problem based on the interrogation.

1        38.    The system of claim 36, wherein the code for detecting the configuration  
2 problem is capable of causing the processor to further perform:  
3        receiving at least one report from at least one component in the network; and  
4        determining whether there is a configuration problem based on the report.

1        39.    The system of claim 36, wherein the code is capable of causing the  
2 processor to further perform:

3       when a component in the network identifies a solution, allowing the component to  
4       automatically apply the solution.

1       40.     The system of claim 36, wherein the configuration problem is at least one  
2       of a network configuration problem and a storage configuration problem.

1       41.     A method for performing configuration checking of a network,  
2       comprising:  
3       scanning a network data store for at least one transaction;  
4       generating at least one event for said transaction;  
5       associating at least one configuration policy with said event;  
6       comparing the said configuration policy with configuration data associated with  
7       said event; and  
8       determining whether said configuration policy has been violated based on the  
9       comparison.

1       42.     A method for performing proactive configuration checking of a network,  
2       comprising:  
3       receiving a hypothetical network scenario;  
4       generating at least one transaction based on said hypothetical network scenario;  
5       populating a network data store with configuration data for said transaction;  
6       generating at least one event for said transaction using a mapping of events to  
7       transactions; and  
8       using configuration data associated with said event to determine whether a  
9       configuration policy has been violated.

1           43.    A method for performing reactive configuration checking of a network,  
2 comprising:  
3           receiving a request to perform configuration checking on an existing network  
4 configuration;  
5           scanning a network data store for at least one transaction;  
6           generating at least one event for said transaction using a mapping of events to  
7 transactions; and  
8           using configuration data associated with said event to determine whether a  
9 configuration policy has been violated.

1           44.    A method for correcting a configuration problem, comprising:  
2           discovering the configuration problem;  
3           determining whether there is at least one solution for the configuration problem in  
4 a knowledge data store;  
5           when it is determined that there is at least one solution in the knowledge data  
6 store, automatically selecting a solution to solve the configuration problem;  
7           when said solution can be automatically applied, automatically applying said  
8 solution; and  
9           when said solution cannot be automatically applied, notifying a user.